L Number	Hits	Search Text	DB	Time stamp
-	142	717/139.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/08 14:46
-	10672	interpret$3 and native and (switch$3 or alternat$3 or call$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:29
-	10715	interpret$3 and native and (switch$3 or alternat$3 or call$3 or invok$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:30
-	98	interpret$3 same native and (switch$3 or alternat$3 or call$3 or invok$3) and (return adj address) and point$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 12:20
-	80	interpret$3 same native and (switch$3 or alternat$3 or call$3 or invok$3) and (return adj address) and point$3 and stack and register	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 13:54
-	13	("4463423" | "5367685" | "5418964" | "5740441" | "5761477" | "5784553" | "5787431" | "5848274" | "5848423" | "5857197" | "5864862" | "5907707" | "5913065").PN.	USPAT	2003/10/07 13:19
-	28	6081665.URPN.	USPAT	2003/10/07 13:24
-	693	interpret$3 and native and (("32" adj bit) or 32-bit) and (8-bit or bytecode or byte-code or ("8" adj bit) or java)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 15:29
-	84	interpret$3 same native and (("32" adj bit) or 32-bit) same (8-bit or bytecode or byte-code or ("8" adj bit) or java)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 16:46
-	7	("5335344" | "5367685" | "5381547" | "5590331" | "5675801" | "5761513" | "5787431").PN.	USPAT	2003/10/07 15:42
-	10	5923878.URPN.	USPAT	2003/10/07 15:46
-	45759	photon	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 16:46
-	163	photon same interpret$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 16:47
-	23	photon same interpret$3 and ARM	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 16:47
-	2	photon same interpret$3 and ARM and native	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 16:47

-	2	photon same interpret$3 and 7??/???.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 16:48
-	135	memory and (native and (bytecode or byte-code)) near3 instruction	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 17:09
-	0	memory and (native and (bytecode or byte-code)) near3 instruction and (memory adj arrangement)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/07 17:09
-	2	6317872.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/08 16:10
-	339	interpret$3 near5 cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 07:29
-	25	interpret$3 near5 (store$3 near3 cache)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:04
-	0	(instantiat$3 near3 interpret$3) near3 cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:05
-	49	dual near5 compil$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:06
-	67	hybrid near5 compil$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:25
-	49	(dual near5 compil$5) not (hybrid near5 compil$5)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:27
-	6	(dual near5 compil$5) not (hybrid near5 compil$5) and interpret$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:27
-	1556	interpret$3 and native and (switch$3 or alternat$3 or call$3) and cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:29
-	756	interpret$3 and native and (switch$3 or alternat$3 or call$3 or invok$3)and instruction and address and register and stack	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:31

-	362	((interpret$3 or vm or (virtual adj machine)) same compile) and (hll or high-level or 32bit or 32-bit or ("32" adj bit) or native) same (bytecode or byte-code or (byte adj code) or 8bit or 8-bit or bit or java)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 08:44
-	134	((interpret$3 or vm or (virtual adj machine)) same compile) and (hll or high-level or 32bit or 32-bit or ("32" adj bit) or native) same (bytecode or byte-code or (byte adj code) or 8bit or 8-bit or bit or java) and address and return and stack and register	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:15
-	40	(cross-platform or (cross adj platform)) near3 code	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:24
-	402	((6158045.PN. and (((interpret$3 or vm or (virtual adj machine)) same compile) and (hll or high-level or 32bit or 32-bit or ("32" adj bit) or native) same (bytecode or byte-code or (byte adj code) or 8bit or 8-bit or bit or java) and address and return and stack and register)) or 32-bit or 32bit) and (java or bytecode or byte-code or byte) and (compil$5 same interpret$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:31
-	747	709/100.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:57
-	520	709/310.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:32
-	454	709/107.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:33
-	543	712/245.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:41
-	247	712/209.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:39
-	0	712/209.ccls. and ((6295642.PN. and (((interpret$3 or vm or (virtual adj machine)) same compile) and (hll or high-level or 32bit or 32-bit or ("32" adj bit) or native) same (bytecode or byte-code or (byte adj code) or 8bit or 8-bit or bit or java) and address and return and stack and register)) and (6158045.PN. and (((interpret$3 or vm or (virtual adj machine)) same compile) and (hll or high-level or 32bit or 32-bit or ("32" adj bit) or native) same (bytecode or byte-code or (byte adj code) or 8bit or 8-bit or bit or java) and address and return and stack and register)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:40
-	1	712/209.ccls. and (hll and bytecode)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:40

-	0	712/245.ccls. and ((6295642.PN. and (((interpret$3 or vm or (virtual adj machine)) same compile) and (hll or high-level or 32bit or 32-bit or ("32" adj bit) or native) same (bytecode or byte-code or (byte adj code) or 8bit or 8-bit or bit or java) and address and return and stack and register)) and (6158045.PN. and (((interpret$3 or vm or (virtual adj machine)) same compile) and (hll or high-level or 32bit or 32-bit or ("32" adj bit) or native) same (bytecode or byte-code or (byte adj code) or 8bit or 8-bit or bit or java) and address and return and stack and register)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:41
-	0	712/245.ccls. and (hll and bytecode)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:41
-	0	712/245.ccls. and (32-bit and bytecode)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:41
-	28	712/245.ccls. and (interpret$5 and compil$5)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:51
-	7	712/245.ccls. and (mix$3 near3 instruction)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:52
-	1	709/100.ccls. and (risc or cisc) same (bytecode or byte-code or java)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:55
-	0	709/100.ccls. and (32bit) same (bytecode or byte-code or java)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:56
-	1	709/100.ccls. and (32-bit) same (bytecode or byte-code or java)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 11:56
-	47	709/100.ccls. and (compil$5 same interpret$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 12:46
-	0	712.300.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 12:46
-	264	712/300.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 12:52
-	247	712/209.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 12:47

-	382	712/244.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 12:47
-	145	cross adj language	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 13:05
-	20	persistent near3 (vm or (virtual adj machine))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 13:28
-	1	z8 and (zylog or zylogue)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 15:30
-	2	5504930.PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/09 15:31
-	2	6513156.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/19 17:00
-	0	6513156.pn. and register	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/19 17:00
-	1	6513156.pn. and stack	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/19 17:00
-	192	(interpreter or vm or (virtual adj machine)) near5 cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/21 13:25
-	131	(interpreter or vm or (virtual adj machine)) near3 cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/21 13:26
-	29	(interpreter or vm or (virtual adj machine)) near3 (stor$3 or locat$3) near3 cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/21 13:50
-	985	(interpreter or vm or (virtual adj machine)) near3 (speed or fast)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/21 13:51
-	186	(interpreter or vm or (virtual adj machine)) near3 (frequency)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/21 13:51

-	19	((interpreter or vm or (virtual adj machine)) near3 (speed or fast)) and ((interpreter or vm or (virtual adj machine)) near3 (frequency))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/21 13:58
-	2	((interpreter or vm or (virtual adj machine)) near3 (speed or fast)) and ((interpreter or vm or (virtual adj machine)) near3 (frequency)) and cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/10/21 13:58

Welcome to IEEE Xplore®
○ Home
○ What Can I Access?
○ Log-out

Tables of Contents
○ Journals & Magazines
○ Conference Proceedings
○ Standards

Search
○ By Author
○ Basic
○ Advanced

Member Services
○ Join IEEE
○ Establish IEEE Web Account
○ Access the IEEE Member Digital Library

Print Format

Your search matched **1** of **976857** documents.

A maximum of **1** results are displayed, **50** to a page, sorted by **Relevance** ir
You may refine your search by editing the current search expression or enter
Then click **Search Again**.

((((interpret*)and (compil*)) and(register and stack)) and ((19

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **Java bytecode to native code translation: the Caffeine prototype and preliminary results**
Hsieh, C.-H.A.; Gyllenhaal, J.C.; Hwu, W.W.;
Microarchitecture, 1996. MICRO-29. Proceedings of the 29th Annual IEEE/ACM International Symposium on , 2-4 Dec. 1996
Page(s): 90 -97

[Abstract] [PDF Full-Text (936 KB)] **IEEE CNF**

Citation

Java Grande Conference >archive
Proceedings of the ACM 1999 conference on Java Grande >toc
1999 , San Francisco, California, United States

Java annotation-aware just-in-time (AJIT) complilation system

Authors

Ana Azevedo
Alex Nicolau
Joe Hummel

Sponsor

SIGPLAN : ACM Special Interest Group on Programming Languages

Publisher

ACM Press New York, NY, USA

Pages: 142 - 151 Series-Proceeding-Article
Year of Publication: 1999
ISBN:1-58113-161-5

doi> http://doi.acm.org/10.1145/304065.304115 (Use this link to Bookmark this page)

> full text > references > citings > index terms > peer to peer

> Discuss > Similar > Review this Article Save to Binder

> BibTex Format

↑ FULL TEXT: _ Access Rules

pdf 1.26 MB

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 R. Graft A. Krall. Efficient JavaVM Just-in-Time Compilation. In Proceedings of International Conference on Parallel Architectures and Compilation Techniques, PACT'98, 1998.

2 Ali-Reza Adl-Tabatabai , Micha? Ciemiak , Guei-Yuan Lueh , Vishesh M. Parikh , James M. Stichnoth, Fast, effective code generation in a just-in-time Java compiler, Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation, p.280-290, June 17-19, 1998, Montreal, Quebec, Canada

3 G. Bilardi and A. Nicolau. Adaptive Bitonic Sorting: An OptimM Parallel Algorithm for Shared Memory Machines. Technical Report TR86-769, Cornell University, 1986.

4 G. J. Chaitin. Register Allocation and Spilling via Graph Coloring. SIGPLAN Notices, 17(6):201- 107, June 1982.

5 G. J. Chaitin, M. A. Auslander, A. K. Chandra, J. Cocke, M. E. Hopkins, and P. W. Markstein. Register Allocation via Coloring. Computer Languages, 6:47-57, January 1981.

6 Fred C. Chow , John L. Hennessy, The priority-based coloring approach to register allocation, ACM Transactions on Programming Languages and Systems (TOPLAS), v.12 n.4, p.501-536, Oct. 1990

7 M. Ciemiak and W. Li. Optimizing Java Bytecodes. Concurrency:Practice and Experience, 9(11), November 1997.

8 L. R. Clausen. A Java Bytecode Optimizer Using Side-effect Analysis. Concurrency: Practice and Experience, 9(11), November 1997.

9 A. Krishnamurthy , D. E. Culler , A. Dusseau , S. C. Goldstein , S. Lumetta , T. von Eicken , K. Yelick, Parallel programming in Split-C, Proceedings of the 1993 ACM/IEEE conference on Supercomputing, p.262-273, December 1993, Portland, Oregon, United States

10 Michael Franz , Thomas Kistler, Slim binaries, Communications of the ACM, v.40 n.12, p.87-94, Dec. 1997

11 James Gosling , Bill Joy , Guy L. Steele, The Java Language Specification, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1996

12 Brian Grant , Markus Mock , Matthai Philipose , Craig Chambers , Susan J. Eggers, Annotation-directed run-time specialization in C, Proceedings of the 1997 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation, p.163-178, June 12-13, 1997, Amsterdam, The Netherlands

13 David Griswold. The Java HotSpot Virtual Machine Architecture, March 1998. See whitepaper at http://www .j avasoft, com/products/hot spot/.

14 Cheng-Hsueh A. Hsieh , John C. Gyllenhaal , Wen-mei W. Hwu, Java bytecode to native code translation: the caffeine prototype and preliminary results, Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture, p.90-99, December 02-04, 1996, Paris, France

15 J. Hummel, A. Azevedo, D. Kolson, and A. Nicolau. Annotating the Java Bytecodes in Support of Optimization. Concurrency: Practice and Experience, 9(11):1003-1016, November 1997.

16 Microsoft Inc. The Microsoft Virtual Machine for Java. See http://www.microsoft.com/java/sdk/.

17 SUN Inc. Sun interpreter. See http://www.javasoft.com.

18 Symantec Inc. Just in Time Compiler for Windows 95/NT. See http://www.symantec.com.

19 T. Kistler and M. Franz. Dynamic Runtime Optimization. In Proceedings of the Joint Modular Languages Conference, JMLC'97, pages 53-66, March 1997.

20 Massimiliano Poletto , Dawson R. Engler , M. Frans Kaashoek, tcc: a system for fast, flexible, and high-level dynamic code generation, Proceedings of the 1997 ACM SIGPLAN conference on Programming language design and implementation, p.109-121, June 16-18, 1997, Las Vegas, Nevada, United States

21 T. Proebsting, J. Hartman, G. Townsend, P. Bridges, T. Newsham, and S. Watterson. Toba: A Java-to-C translator. See http://www.cs.arizona.edu/sumatra/toba.

22 Effective Edge Technologies. guavac. See summit.stanford.edu:/pub/guavac/.

23 David W. Wall, Global register allocation at link time, Proceedings of the SIGPLAN symposium on Compiler contruction, p.264-275, June 25-27, 1986, Palo Alto, California, United States

24 Tim Wilkinson. Kaffe: A Free JIT virtual machine to run Java code. See http://www.transvirtual.com.

↑ CITINGS *11*

Patrice Pominville , Feng Qian , Raja Vallée-Rai , Laurie Hendren , Clark Verbrugge, A framework for optimizing Java using attributes, Proceedings of the 2000 conference of the Centre for Advanced Studies on Collaborative research, p.8, November 13-16, 2000, Mississauga, Ontario, Canada

Ayal Zaks , Vitaly Feldman , Nava Aizikowitz, Sealed calls in Java packages, ACM SIGPLAN Notices, v.35 n.10, p.83-92, Oct. 2000

Jagun Kwon , Andy Wellings , Steve King, Ravenscar-Java: a high integrity profile for real-time Java, Proceedings of the 2002 joint ACM-ISCOPE conference on Java Grande, p.131-140, November 03-05, 2002, Seattle, Washington, USA

Igor Pechtchanski , Vivek Sarkar, Immutability specification and its applications, Proceedings of the 2002 joint ACM-ISCOPE conference on Java Grande, p.202-211, November 03-05, 2002, Seattle, Washington, USA

Michael Thies, Annotating Java libraries in support of whole-program optimization, Proceedings of the inaugural conference on the Principles and Practice of programming, 2002 and Proceedings of the second workshop on Intermediate representation engineering for virtual machines, 2002, June 13-14, 2002, Dublin, Ireland

Chandra Krintz , Brad Calder, Using annotations to reduce dynamic optimization time, ACM SIGPLAN Notices, v.36 n.5, p.156-167, May 2001

Raja Vallée-Rai , Phong Co , Etienne Gagnon , Laurie Hendren , Patrick Lam , Vijay Sundaresan, Soot - a Java bytecode optimization framework, Proceedings of the 1999 conference of the Centre for Advanced Studies on Collaborative research, p.13, November 08-11, 1999, Mississauga, Ontario, Canada

Iffat H. Kazi , Howard H. Chen , Berdenia Stanley , David J. Lilja, Techniques for obtaining high performance in Java programs, ACM Computing Surveys (CSUR), v.32 n.3, p.213-240, Sept. 2000

Chandra Krintz, Coupling on-line and off-line profile information to improve program performance, Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization, p.69, March 23-26, 2003, San Francisco, California

Jagun Kwon , Andy Wellings , Steve King, Assessment of the Java programming language for use in high integrity systems, ACM SIGPLAN Notices, v.38 n.4, April 2003

John Aycock, A brief history of just-in-time, ACM Computing Surveys (CSUR), v.35 n.2, p.97-113, June 2003

↑ INDEX TERMS

Primary Classification:
D. Software
↳ **D.3** PROGRAMMING LANGUAGES

Additional Classification:
D. Software
↳ **D.2** SOFTWARE ENGINEERING

General Terms:
Design, Measurement

↑ Peer to Peer - Readers of this Article have also read:

- We Talk to Everybody
  Linux Journal 2000, 74es
  Marjorie Richardson , Jason Schumaker , David Penn

- Editorial pointers
  Communications of the ACM 44, 9
  Diane Crawford

- News track
  Communications of the ACM 44, 9
  Robert Fox

- At the Forge
  Linux Journal 1998, 52es
  Reuven M. Lerner

- Forum

Communications of the ACM 44, 9
Diane Crawford

PORTAL

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design

Give us your opinion after using it.

Search Results

Search Results for: **[interpret* and native and stack and register]**
Found **369** of **121,820 searched.**

Warning: Maximum result set of 200 exceeded. Consider refining.

Search within Results

GO

> Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score

Results 1 - 20 of 200 short listing

Prev Page 1 2 3 4 5 6 7 8 9 10 Next Page

1 Techniques for obtaining high performance in Java programs 99%

Iffat H. Kazi , Howard H. Chen , Berdenia Stanley , David J. Lilja
ACM Computing Surveys (CSUR) September 2000
Volume 32 Issue 3

This survey describes research directions in techniques to improve the performance of programs written in the Java programming language. The standard technique for Java execution is interpretation, which provides for extensive portability of programs. A Java interpreter dynamically executes Java bytecodes, which comprise the instruction set of the Java Virtual Machine (JVM). Execution time performance of Java programs can be improved through compilation, possibly at the expense of portabili ...